Reducts Versus Constructs: an Experimental Evaluation

Robert Susmaga [1]

Institute of Computing Science, Poznań University of Technology
60-965 Poznań, Poland

Abstract

A reduct is one of the main notions in the Rough Sets Theory (RST). The idea of the reduct proved to be interesting enough to inspire a great deal of research resulting in various reduct-related notions and concepts. Reducts have been the subject of multiple papers, in which various approaches to the problem of defining, generating and applying reducts have been proposed. Given a data set, in which some pre-defined objects are described in terms of numerous parameters, the idea is to obtain a more compact description of these objects. As there are no universal descriptions for all possible applications, the idea is to focus on a description that is advantageous from a particular point of view, e.g. represented in the form of a consistency condition. Such a description may be created when sparse/redundant attributes are identified and eliminated from the data. In its classic form, reducts are minimal subsets of attributes that retain the consistency condition. The paper introduces the idea of constructs, in which the condition is modified so that the construct should manifest even better properties than those of reducts.

1 Introduction

All data analyses in the Rough Sets Theory [8] start with the so-called closed-world assumption. According to this assumption any two objects described by two identical vectors of parameter values must be treated equal in all the subsequent analyses. Formally, the main tool that ensures this property in data analysis is the relation of indiscernibility between objects. Thus, any non-empty subset of discrete parameters (attributes) induces a partition of objects into subsets having the same values of all considered attributes [8].

Although the relation of indiscernibility is defined for any non-empty subset of attributes, particular attention is put to partitions induced by the sets of all condition and decision attributes. This allows for distinction between

[1] Email: Robert.Susmaga@cs.put.poznan.pl

premises and conclusions in the analyses. The attributes treated as the decision ones induce a partition of objects into subsets that are referred to as classes (usually only one such attribute is taken into consideration at a time). The partition of objects into classes is very interesting to the data analysts, because the classes represent concepts that could be later analyzed and described in terms of the condition information. The condition attributes, on the other hand, represent all the working properties of the objects.

Having defined the partitions, induced among objects by indiscernibility defined by the sets of condition and decision attributes, the RST makes an effort to examine whether the set of condition attributes is sufficient to classify objects into elements of the partition induced by the set of decision attributes. The theory introduces a specialized measure, referred to as the quality of approximation of objects' classification [8], which represents the percentage of objects that may be classified univocally into appropriate classes induced by the decision attributes. If all objects are classified correctly, the quality is equal to 1.0. Lower values of this measure, on the other hand, indicate existence of inconsistent objects (ones described by the same values of condition attributes but different values of the decision ones). Controlled handling of inconsistency is important as it may be easily introduced into consistent data when performing various forms of preprocessing, e.g. during the discretization process.

Introduction of the quality of approximation allowed the RST to proceed with further analyses, in particular with the problem of attribute redundancy. An attribute may be treated as redundant if its removal does not change the value of the quality of approximation. A subset of attributes that does not contain redundant attributes may be referred to as non-redundant. A non-redundant set of attributes for which the quality is not lower than for the set of all attributes is called a reduct [8]. Thus a given reduct contains attributes that are (in a way) more important than those that are not included in the reduct. Generating reducts is then a form of attribute evaluation.

Obviously, multiple reducts for one set of attributes may exist and searching for reducts need not be an easy task. In fact, the process of searching for all reducts may be proved to be NP–hard [13]. In spite of the complexity, a great effort had been exercised to discover methods of effective reduct generation. This concerns exact [4, 11, 13, 17] as well as approximate approaches [12, 14, 18].

Results of many papers confirm the fact that data analyses, including reduct generation schemes based on the quality of approximation of objects' classification can be successfully applied in different areas of human interest [5, 15]. The measure has played a special role in the RST as the main consistency measure of the theory.

The very idea of attribute reduction is probably most closely related to that of Feature Selection, which also has been studied thoroughly in numerous papers [1, 3]. Application of reducts in similar role has been examined in many papers on the Rough Set Theory, where reduction of attributes remains one of the main issues, as well as in numerous papers concerning various aspects of information systems analysis [9, 13].

The rest of the paper is organized as follows. Section 2 presents the basic definitions. Section 3 introduces the idea of constructs. Section 4 presents the experimental evaluation and the last section contains final conclusions and remarks.

2 Terminology and Basic Definitions

The main data set considered in this paper is a *decision table*, which is a special case of an *information table* [8]. Formally the decision table is defined as a 4-tuple $DT=\langle U, Q, V, \delta \rangle$, where:

- U is a non-empty, finite set of objects under consideration,
- Q is a non-empty, finite set of condition ($C\neq\emptyset$) and decision ($D\neq\emptyset$) attributes, such that $C \cup D = Q$ and $C \cap D = \emptyset$; in this paper it will be further assumed that $D=\{d\}$,
- V is a non-empty, finite set of attribute values,
- δ is an information function, δ: $U \times Q \to V$.

A differently formulated, but equivalent definition of the information systems may be found e.g. in [13].

Let $IND(P) \in U \times U$ denote an *indiscernibility relation*, defined for a non-empty set of attributes $P \subseteq Q$ as:

$$IND(P) = \{(x,y) \in U \times U\colon \forall_{q \in P} \quad \delta(x,q) = \delta(y,q)\}.$$

If a pair of objects belongs to $IND(P)$ then these two objects are indiscernible from each other on all attributes from the set P. The relation $IND(P)$ is reflexive, symmetric and transitive (it is an equivalence relation).

By $DIS(P) \in U \times U$, the *discernibility relation*, we shall denote the opposite relation, defined as:

$$DIS(P) = \{p \in U \times U\colon\ p \notin IND(P)\}.$$

If a pair of objects belongs to $DIS(P)$ then these two objects differ on at least one attribute from the set P. The relation $DIS(P)$ is not reflexive and not transitive, but it is symmetric.

Finally, let *SIM(P)*$\in U \times U$ denote a *similarity relation*, defined for a set of attributes $P \subseteq Q$ as:

$$SIM(P) = \{(x,y) \in U \times U \colon \underset{q \in P}{\exists} \quad \delta(x,q) = \delta(y,q)\}.$$

If a pair of objects belongs to *SIM(P)* then these two objects are indiscernible on at least one attribute from the set P. In other words $(x,y) \in SIM(P)$ iff $\underset{\emptyset \neq P' \subseteq P}{\exists} (x,y) \in IND(P')$. This particular relation $SIM(P)$ is reflexive and symmetric, but it is not transitive (it is a tolerance relation).

Given $P \subseteq Q$, the relation $IND(P)$ induces a partition of objects into subsets. In particular, $IND(\{d\})$ partitions the objects in to subsets referred to as classes. Thus if $(x,y) \in IND(\{d\})$ then the objects x and y are said to belong to the same class; otherwise they are said to belong to different classes.

According to its classic definition, the idea of a *relative reduct* is to distinguish objects belonging to different classes. If a subset of condition attributes is to satisfy the definition of a relative reduct, it has to be able to distinguish all those objects belonging to different classes that are also distinguished by the whole set of condition attributes. It should be stressed that in what follows the idea of distinguishing objects belonging to different classes is mentioned very often. In fact, due to potential inconsistency in data, we are only capable of distinguishing objects belonging to lower approximations of classes [8] and this is what will be implied, even if not stated, throughout the paper.

The formal definition (equivalent to the classic definition of a relative reduct, [13]), but assuming an object-oriented point of view, is as follows.

A subset of condition attributes R ($R \subseteq C$) is a *relative reduct* iff:

(F1) $\quad \underset{p \in U \times U}{\forall} \{[p \in DIS(D) \wedge p \in DIS(C)] \rightarrow p \in DIS(R)\}$

(F2) $\quad \underset{q \in R}{\forall} \underset{p \in U \times U}{\exists} \{p \in DIS(D) \wedge p \in DIS(R) \wedge p \notin DIS(R - \{q\})\}$

The first formula ensures that the reduct has not lower ability to distinguish objects belonging to different classes than the whole set of attributes (this feature may be referred to as *consistency*). The second requires that the reduct is minimal with regard to inclusion, i. e. it does not contain redundant attributes or, in other words, it does not include other reducts (further referred to as *minimality*).

It should be also noted that minimality with regard to inclusion does not imply minimality with regard to set cardinality. In fact, reducts generated for a given decision table may vary considerably as far as their cardinality is concerned.

3 Constructs: Combination of Inter-class and Intra-class Reducts

It is assumed that the definition of relative reducts given in form of formulae F1 and F2 defines what will be further referred to as *inter-class reducts*, i. e. subsets of attributes that ensure sufficient discernibility of objects belonging to different classes.

Assuming point of view results in defining intra-class reducts, i.e. subset of attributes that ensure similarity between objects belonging to the same class. The formal definition is as follows.

A subset of condition attributes R $(R \subseteq C)$ is an *intra-class reduct* iff:

$$\text{(F3)} \qquad \forall_{p \in U \times U} \{[p \in SIM(D) \land p \in SIM(C)] \rightarrow p \in SIM(R)\}$$

$$\text{(F4)} \qquad \forall_{q \in R} \exists_{p \in U \times U} \{p \in SIM(D) \land p \in SIM(R) \land p \notin SIM(R - \{q\})\}$$

These definitions bear close resemblance to those given by formulae F1 and F2. An intra-class reduct is a set of attributes that allows finding a similarity between every pair of objects belonging to the same class. This is guaranteed by the formula F3. Formula F4 ensures the reduct's minimality with regard to inclusion.

Combinations of inter-class and intra-class reducts seem interesting because the resulting subset of condition attributes would ensure not only the ability to distinguish objects belonging to different classes, but also similarity between objects belonging to the same class. Such a subset could prove very useful when applied with some induction-based methods in further analyses.

Because it is unlikely that the subsets of condition attributes satisfying the formulae F1 and F2 (i.e. the definition of inter-class reducts) would also satisfy the formulae F3 and F4) (i.e. the definition of intra-class reducts) and the other way round, two simple methods of generating combined inter-class and intra-class reducts might be introduced:

- Generate all inter-class reducts (formulae F1 and F2) and discard those not satisfying F3.
- Generate all intra-class reducts (formulae F3 and F4) and discard those not satisfying F1.

The result, in both cases, is a potentially empty set of subsets of condition attributes, with each of the subsets satisfying both F1 and F3, i.e. ensuring discernibility between all objects belonging to different classes and similarity between all objects belonging to the same classes. What is not properly solved in this case is the problem of minimality: the resulting subsets of attributes are not *all* the minimal (with regard to inclusion) subsets satisfying both F1 and F3. That is because the above methods can be thought of as eliminative. An alternative, constructive method of defining all such subsets is as follows.

A subset of condition attributes R ($R \subseteq C$) is a *construct* iff:

(F5) $$\underset{p\in U\times U}{\forall}\{[p \in DIS(D) \wedge p \in DIS(C)] \Rightarrow p \in DIS(R)\}$$

(F6) $$\underset{p\in U\times U}{\forall}\{[p \in SIM(D) \wedge p \in SIM(C)] \Rightarrow p \in SIM(R)\}$$

(F7) $$\underset{q\in R}{\forall}\begin{cases}\underset{p\in U\times U}{\exists}\{p \in DIS(D) \wedge p \in DIS(R) \wedge p \notin DIS(R-\{q\})\} \\ \qquad\qquad\qquad or \\ \underset{p\in U\times U}{\exists}\{p \in SIM(D) \wedge p \in SIM(R) \wedge p \notin SIM(R-\{q\})\}\end{cases}$$

So, a construct is a subset of attributes that retains the discernibility of objects belonging to different classes as well as the similarity of objects belonging to the same class (formulae F5 and F6). Alike reduct, the construct R is minimal, which means that removing any attribute from R would result in making the any (or both) of the conditions given by F5 or F6 invalid.

Because the notion of constructs is similar to that of reducts, the constructs can be generated using a properly modification algorithm for generating reducts. In this paper a modification of the Fast Reduct Generating Algorithm (FRGA, [16]) was applied.

4 Reducts versus Constructs – An Experimental Evaluation

The data sets used in the conducted experiments are real-life data sets of different origin. The sets had been created for scientific purposes and had been previously used in different experiments and analyses. Those analyses were, however, not always related to the problem of reduct/construct generation.

The following sets: Lsd (Large Soybean Database) and Mushroom (Mushroom Database) come from the Irvine Repository of Machine Learning Databases [6]. The sets: Livdpl and Lymph are medical data set obtained from other sources [15]. Mush_010, Mush_020 and Mush_030 are modifications of Mushroom. The idea of modifications was as follows: Mush_010 is the set Mushroom, in which 0.1% of all its values were randomly distorted (the original value was replaced with another value out of the appropriate domain). In Mush_020 twice as much values (0.2%) were distorted, and so on.

The important issue concerning the data sets used in computation of reducts/construct is that they are computable only for discrete condition attributes. Therefore those the continuous attributes of the data sets had to undergo a process of discretization [2, 7]. In all cases a single decision attribute was considered. It is important that the sets are not trivial in the sense that each of them produces at least several hundred of reducts/constructs.
Table 1 contains the basic characteristics of the data sets, which include the

number of condition attributes, objects and classes (distinct values of the decision attribute).

Data Set	#Cond. Attr.	#Objects	#Classes	Comment
Livdpl	22	80	2	original file
Lsd	35	265	15	original file
Lymph	18	148	4	original file
Mushroom	21	8124	2	original file
Mush_010	21	8124	2	distorted Mushroom
Mush_020	21	8124	2	distorted Mushroom
Mush_030	21	8124	2	distorted Mushroom

Table 1
Basic characteristics of the data sets used in the experiments

The proceeding of the experiment was as follows. For each data set a full set of reducts/constructs was initially found. Then, each reduct/construct in turn was applied to reduce the number of attributes in the data set and the reduced data set underwent a re-classification test. The re-classification test was implemented as a k-fold cross-validation. In this kind of test the set of objects is successively split into training and testing samples, which are used to train and test a given classifier. Two classifiers of the *c4.5* family [10] were actually used:

- *c4.5* – inducer of decision trees,
- *c4.5rules* – inducer of decision rules.

Trying to discriminate objects belonging to different classes and to unify objects belonging to the same class is also the main idea in symbolic induction of decision rules. Therefore it seemed reasonable to use a rule generating classifier in the re-classification tests. The actual tool was the rule inducer *c4.5rules*. It starts with generating a decision tree (for which it employs the *c4.5* tree inducer) and then converts the tree to rules by traversing all the root-to-leaf paths in the tree and eliminating redundant conditions in those paths.

This procedure, however, may be quite time-consuming in case of voluminous data sets, which is the case of Mushroom and its modifications. Accordingly, the data sets Mush_010, Mush_020 and Mush_020 (and also Mushroom) were experimented with using only the *c4.5* tree inducer, which is much more quicker than *c4.5rules*.

4.1 Experiments with original data sets

The first batch of the experiments involved four original, real-life data sets – Livdpl, Lsd, Lymph and Mushroom. The presented results include:

- Counts of reduct/construct and average errors by the rule classifier – Table 2.
- Basic statistics (min, mean, etc.) of reducts/constructs in terms of their cardinality – Table 3.

Data Set	Reducts		Constructs	
	Count	Avg rule error	Count	Avg rule error
Livdpl	1295	23.95%	997	23.88%
Lsd	6509	25.53%	6536	25.60%
Lymph	424	25.00%	827	23.95%
Mushroom	572	00.05%	635	00.04%

Table 2
Counts of reduct/construct and average errors by the rule classifier

Data Set	Reduct / Construct Cardinality				
	Min	Mean	Median	Mode	Max
Livdpl	7 / 8	9.7 / 10.2	10 / 10	10 / 10	12 / 13
Lsd	9 / 9	12.5 / 12.5	12 / 12	12 / 12	16 / 16
Lymph	6 / 8	8.4 / 9.9	8 / 10	8 / 10	11 / 12
Mushroom	4 / 5	6.5 / 7.9	7 / 8	7 / 8	8 / 10

Table 3
Basic statistics of reducts/constructs in terms of their cardinality

The general observations are as follows:

- the analyzed data sets tend to produce more constructs than reducts,
- the constructs tend to contain more attributes than the reducts,
- the overall error achieved on constructs is slightly smaller than that achieved on reducts, but the differences are *not* statistically important – the resulting probability in the Student two-tailed test exceeds the 0.01 significance level.

The final remark here is that despite some differences neither reducts nor constructs have any noteworthy advantage over the other option. The numbers of generated reducts and constructs remain large enough to make the process of selecting the best one fairly difficult. At this point it is important to state that from the practical point of view the mere number of generated reducts is an informative indicator of the quality of regularities discovered in data. As long as the number of all reducts is small then the regularities might be strong. On the other hand, when this number becomes big then the generated reducts are often of poor quality, as they simply tend to be all those combinations of attributes that happen to satisfy the definition of reducts. Of course it is still possible that some of these reducts are actually good, but identifying those remains difficult due to their large number.

4.2 Experiments with noise-distorted data sets

The second group of results concerns modified data sets (all modification applied to the original data set Mushroom). It was carried out to verify the assumption that constructs become more useful when excessive noise distorts proper definition of classes. The presented results include:

- Counts of reduct/construct and average errors by the tree classifier – Table 4.
- Basic statistics (min, mean, etc.) of reducts/constructs in terms of their cardinality – Table 5.

Data Set	Reducts		Constructs	
	Count	Avg tree error	Count	Avg tree error
Mushroom	572	0,01%	635	0,00%
Mush_010	1628	0.56%	853	0.45%
Mush_020	1756	1.33%	139	1.28%
Mush_030	1666	1.60%	194	1.34%

Table 4
Counts of reduct/construct and average errors by the tree classifier

Data Set	Reduct / Construct Cardinality				
	Min	Mean	Median	Mode	Max
Mushroom	4 / 5	6.5 / 7.9	7 / 8	7 / 8	8 / 10
Mush_010	6 / 9	8.3 / 11.5	8 / 11	8 / 11	12 / 14
Mush_020	6 / 12	9.6 / 13.7	10 / 14	9 / 14	14 / 16
Mush_030	8 / 13	9.9 / 14.1	10 / 14	10 / 14	13 / 16

Table 5
Basic statistics of reducts/constructs in terms of their cardinality

In data sets with easily separated classes there seems to be no problem in performing reduction using a reduct, as its fundamental task is finding a possibly small subsets attributes that guarantee discernibility between objects belonging to different classes. In noisy environments, however, the problem is different, as the noise makes very many objects differ, also including objects belonging to the same class. In such a case it is very easy to generate a set of attributes that constitutes a legitimate reduct, because the attributes have additional distinctive powers (the powers, however, are not inherent characteristics of the attributes, instead, they originate from the noise-stimulated diversity of the objects). This particular set, however, will discern not only objects belonging to different classes, but also objects belonging to the same class. In result, any subsequent induction-based analyses could produce inferior results.

The general observations in this experiment are as follows:

- the data sets tend to produce notably more reducts than constructs,
- the constructs tend to contain more attributes than the reducts,
- the overall error achieved on constructs is *significantly* smaller than that achieved on reducts; the resulting probability in the Student two-tailed test stays well below the 0.01 significance level.

The results confirm the anticipation: as the amount of noise in the data set grows, the reducts become appreciably shorter (as far as the number of attributes contained in them is concerned). At the same time they become increasingly numerous, i.e. now there are much more reducts than constructs. Finally, the classification errors generated for reducts is significantly higher than that of constructs. All this seems to confirm the fact that constructs constitute much more useful subsets of attributes than reducts.

5 Conclusions

The main purpose of the research reported in this paper has been a presentation of a novel method of attribute reduction, which involves modyfying the classic definition of reducts. The result of this modification is the definition of constructs.

In many aspects the constructs bear close resemblance to the reducts. Constructs have a similar definition and may be generated using a similar algorithm. In some situations, however, they have an advantage over the reducts. Similarly to reducts they discern objects belonging to different classes but, at the same time, they minimize the influence of noise, by trying to retain similarities between objects belonging to the same class. In very noisy environments, often found in real-life applications, the constructs may perform better than the original reducts.

Summarizing, the idea of the construct constitutes a good alternative to that of the reduct. Experiments demonstrate that applying constructs instead of reducts may prove advantageous with noisy data. At the same time, constructs do not produce noticeably worse results than reducts, irrespective of the level of noise.

6 Acknowledgements

The experiments reported in this paper were conducted at the Supercomputing and Networking Center of Poznan, Poland. This paper has been supported by the research grant no. 8T11F-006-19 from the Polish Research Committee (KBN).

References

[1] Dash M., Liu H.: 'Feature Selection for Classification', *Intelligent Data Analysis* (on-line journal), **1** no. 3 (1997), http://www-east.elsevier.com/ida.

[2] Dougherty J. Kohavi R., Sahami M.: 'Supervised and Unsupervised Discretizations of Continuous Features', In: *Proceedings of the 12th International Conference on Machine Learning* Morgan Kaufmann Publishers (1995), 194–202.

[3] John H.G., Kohavi R., Pfleger K.: 'Irrelevant Features and the Subset Selection Problem', In: Cohen W.W. and Hirsh H., (eds), *Machine Learning: Proceedings of the Eleventh International Conference*, Morgan Kaufmann Publishers, San Francisco, CA (1994), 121–129.

[4] Kryszkiewicz M., Rybinski H.: 'Finding Reducts in Composed Information Systems', *Fundamenta Informaticae*, **2**, no. 2–3 (1996), 183–196.

[5] Modrzejewski M.: 'Feature selection using rough sets theory', In: Brazdil P.B., (ed.), *Proceedings of the European Conference on Machine Learning*, (1993), 213–226.

[6] Murphy P.M., Aha D.W.: 'UCI Repository of Machine Learning Databases', University of California, Department of Information and Computer Science, Irvine, CA (1992), WWW page: http://www.ics.uci.edu/~mlearn, e-mail: ml-repository@ics.uci.edu.

[7] Nguyen, H. S., Skowron A.: 'Quantization of Real Value Attributes. Rough Set and Boolean Reasoning Approaches', In: *Proceedings of the Second Annual Join Conference on Information Sciences*, September/October 1995, Wrightsville Beach, NC (1995), 34–37.

[8] Pawlak Z. *Rough Sets. Theoretical Aspects of Reasoning About Data*, Kluwer Academic Publishers, Dordrecht, (1991).

[9] Pawlak Z., Slowinski R.: 'Rough Set Approach to Multi-Attribute Decision Analysis', *European Journal of Operational Research*, **72**, (1994), 443–459.

[10] Quinlan J.R.: *C4.5: Programs for Machine Learning* Morgan Kaufmann, CA, (1993).

[11] Romanski S.: 'Operations on Families of Sets for Exhaustive Search Given a Monotonic Function', In: Beeri, C., Smith, J.W., Dayal, U., (eds), *Proceedings of the 3rd International Conference on Data and Knowledge Bases*, Jerusalem, Israel (1998), 28–30.

[12] Skowron A.: 'Extracting Laws from Decision Tables: A Rough Set Approach', *Computational Intelligence*, **11**, no. 2, (1995), 371–388.

[13] Skowron A., Rauszer C.: 'The Discernibility Functions Matrices and Functions in Information Systems', In: Slowinski R., (ed.), *Intelligent Decision Support. Handbook of Applications and Advanes of the Rough Set Theory*, Kluwer Academic Publishers, Dordrecht (1992), 331–362.

[14] Slezak D.: 'Searching for Frequential Reducts in Decision Tables with Uncertain Objects', In: Polkowski L., Skowron A., (eds), *Proceedings of the First International Conference on Rough Sets and Current Trends in Computing*, Warsaw 1998, Springer-Verlag, Berlin (1998), 52–59.

[15] Slowinski. R., Stefanowski J., Antczak A., Kwias Z.: 'Rough set approach to the verification of indications for treatment of urinary stones by extracorporeal shock wave lithotropsy (ESWL)', In: Lin T.Y, Wildberg A.M., (eds) *Soft Computing*, Society for Computer Simulation, San Diego, California (1995), 93–96.

[16] Susmaga R.: 'Experiments in Incremental Computation of Reducts', In: Skowron A., Polkowski L., (eds), *Rough Sets in Data Mining and Knowledge Discovery*, Springer-Verlag, Berlin (1998a).

[17] Susmaga R.: 'Effective Tests for Minimality in Reduct Generation', *Foundations of Computing and Decision Sciences*, Vol. 23, No. 4, Poznan, Poland (1998d), 219–240.

[18] Wroblewski, J.: 'Finding Minimal Reducts Using Genetic Algorithms', In: *Proceedings of the Second Annual Join Conference on Information Sciences*, September/October 1995, Wrightsville Beach, NC (1995), 186–189.